Machine Learning for Probabilistic Modelling

Iain Murray

School of Informatics, University of Edinburgh

Hello! These slides were visual aids for a talk, and weren't designed to be read. I've inserted some notes here to summarize what the points were supposed to be, and to give further references.

1. Hierarchical modelling is essential. Many models contain large numbers of 'nuisance variables'. We need to *learn* how these are distributed, because if we make assumptions (including vague or so-called 'uninformative' ones), we'll simply get wrong answers. The model I discussed was referring to: "Inferring the force law in the solar-system from a snapshot", Bovy et al., 2010.
`http://arxiv.org/abs/0903.5308`

More thoughts and references on hierarchical modelling are in my discussion `http://homepages.inf.ed.ac.uk/imurray2/pub/11catchup/catchup.pdf` of `http://dx.doi.org/10.1111/j.1467-9868.2011.01025.x`

Hierarchical models can be hard to infer. Example:
`http://homepages.inf.ed.ac.uk/imurray2/pub/10hypers/`

2. Real-world models have a lot of messy detail:
1) complicated instrument-error distributions we may not care about;
2) theory encoded in expensive-to-run simulations.

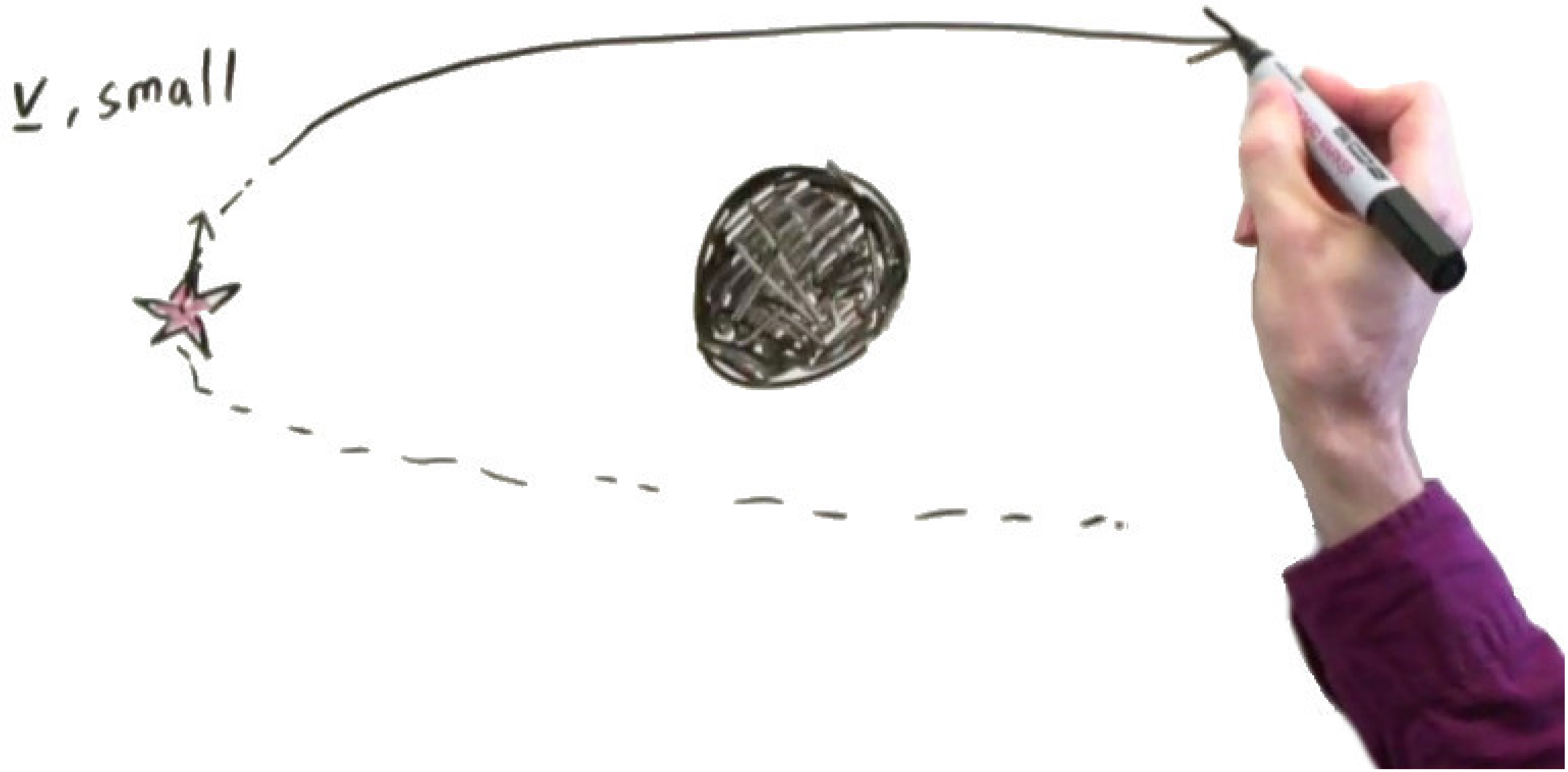
Machine Learning can help.
If we don't want wrong models (point 1.) we need to learn from large amounts of data about our instruments, and from simulation data describing our theories. I've been involved in a series of papers on flexible black-box probabilistic models that could be used here:
`http://homepages.inf.ed.ac.uk/imurray2/pub/11nade/`
`http://homepages.inf.ed.ac.uk/imurray2/pub/13rnade/`
`http://homepages.inf.ed.ac.uk/imurray2/pub/14dnade/`

4. Probabilistic inference methods need extending.
Approximate inference is a heavily-mined and active area. Getting up-to-speed and finding a niche is challenging. However, work in this area is important. In deep and wide graph structures, with billions of observations in some of the plates, it's hard to do fully Bayesian inference.

Some of my work has been on identifying common small inference problems, which are usually only part of an analysis, and developing easier-to-use inference methods for them. E.g.
`http://homepages.inf.ed.ac.uk/imurray2/pub/10ess/`

I'm now also interested in developing easier-to-use methods to summarize and communicate the results of local inferences across large models. I believe the way forward is fitting flexible representations of beliefs, by combining machine learning methods and approximate inference algorithms.

An Inference Problem



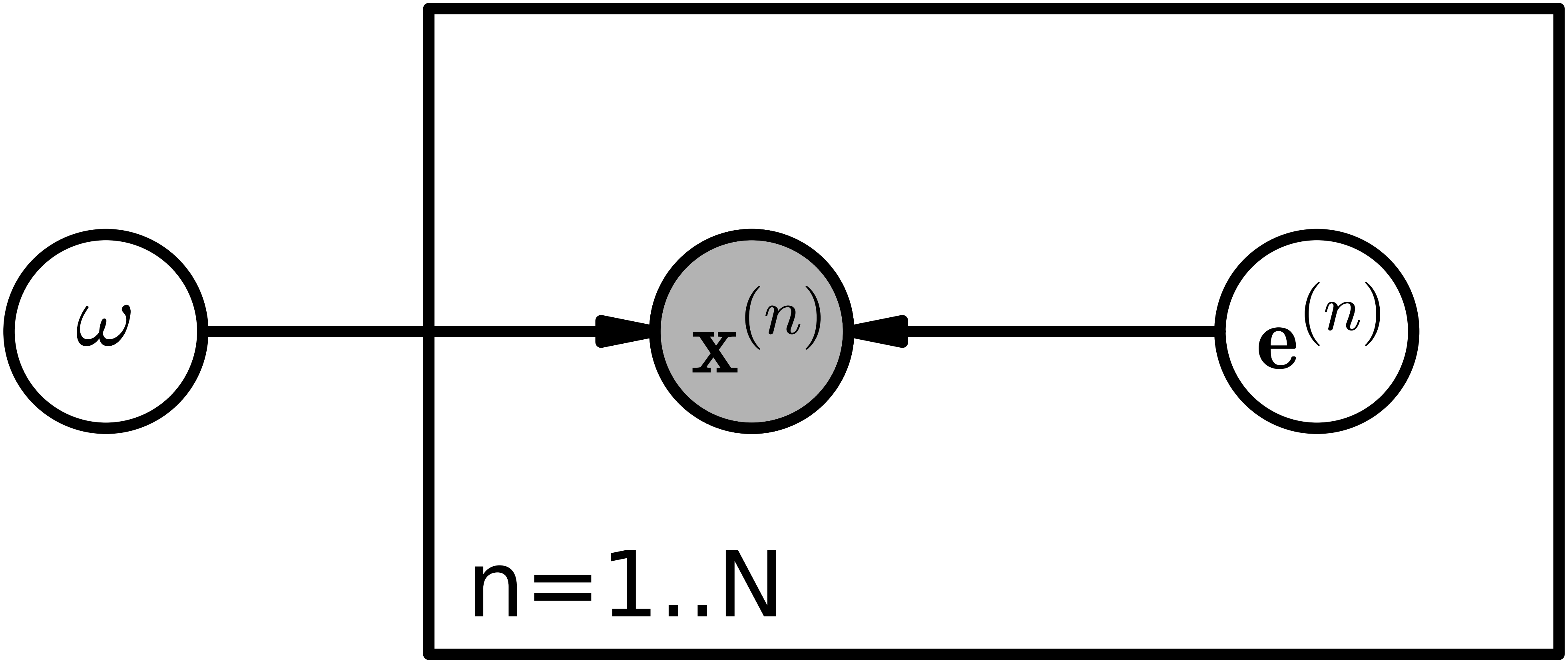
Acceleration law around the sun

$$a(r) = -\boldsymbol{A}\left(\frac{r}{r_0}\right)^{-\alpha}$$

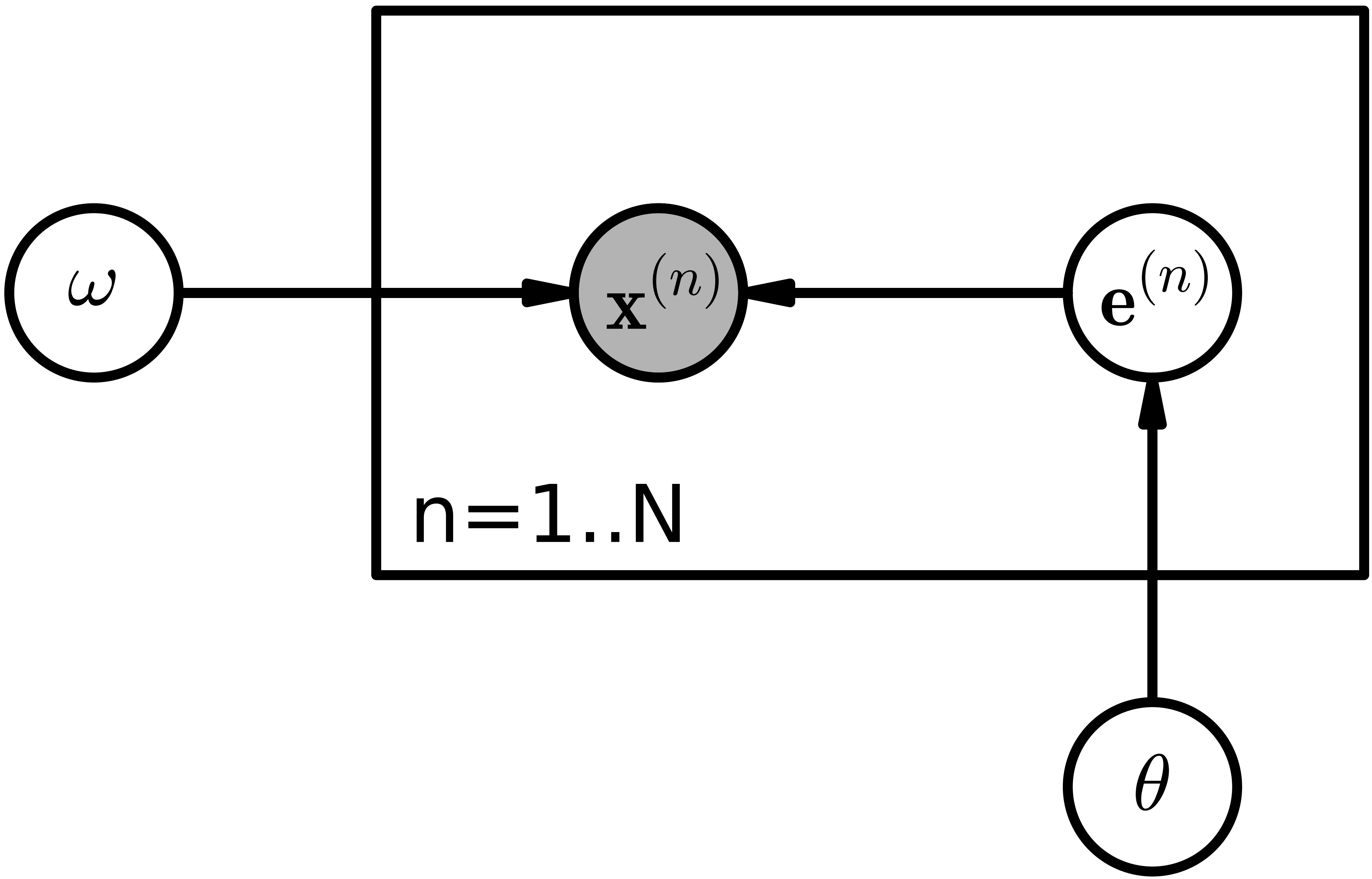
From a snapshot:

8 planet positions and velocities

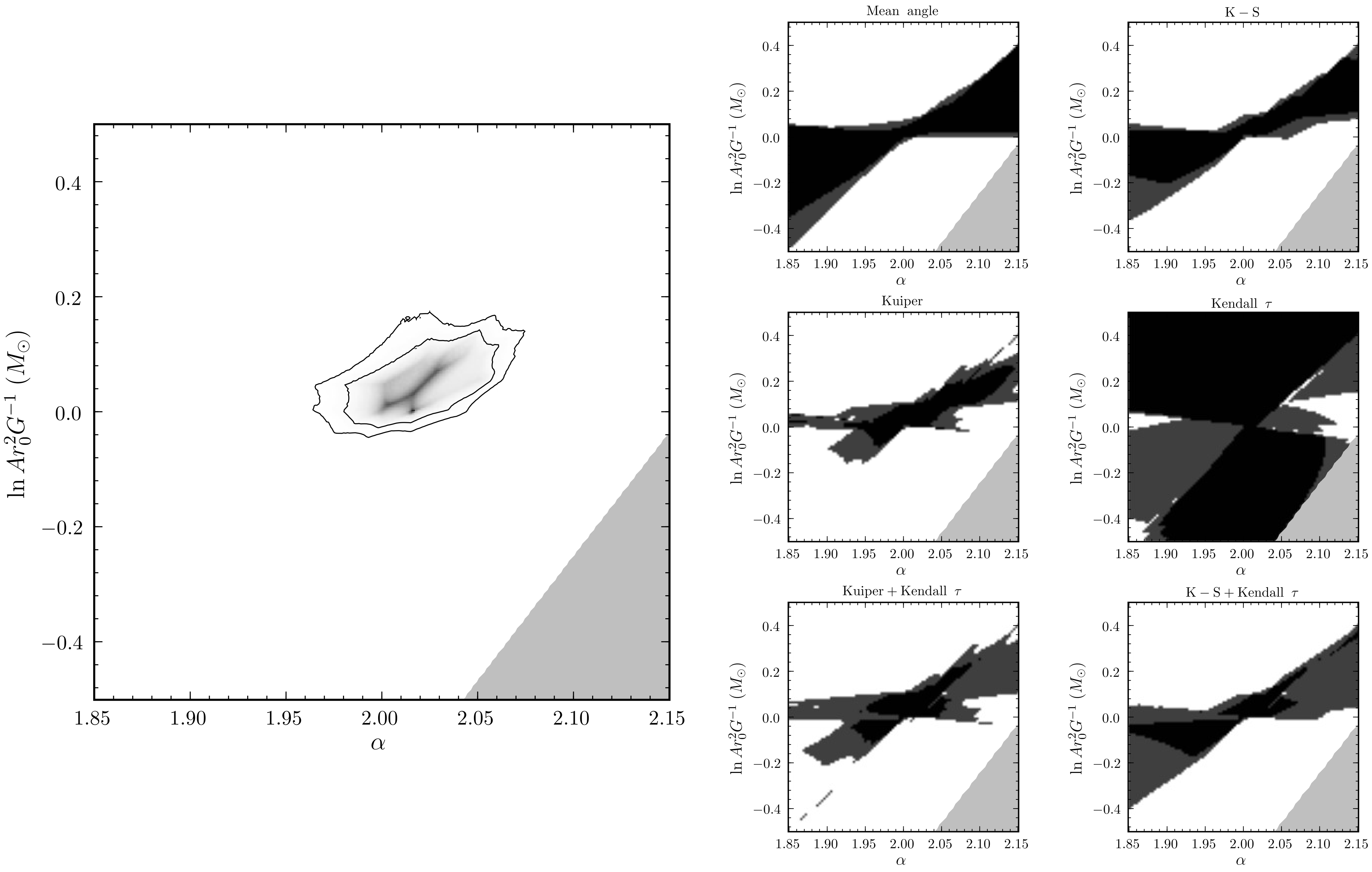
Graphical model



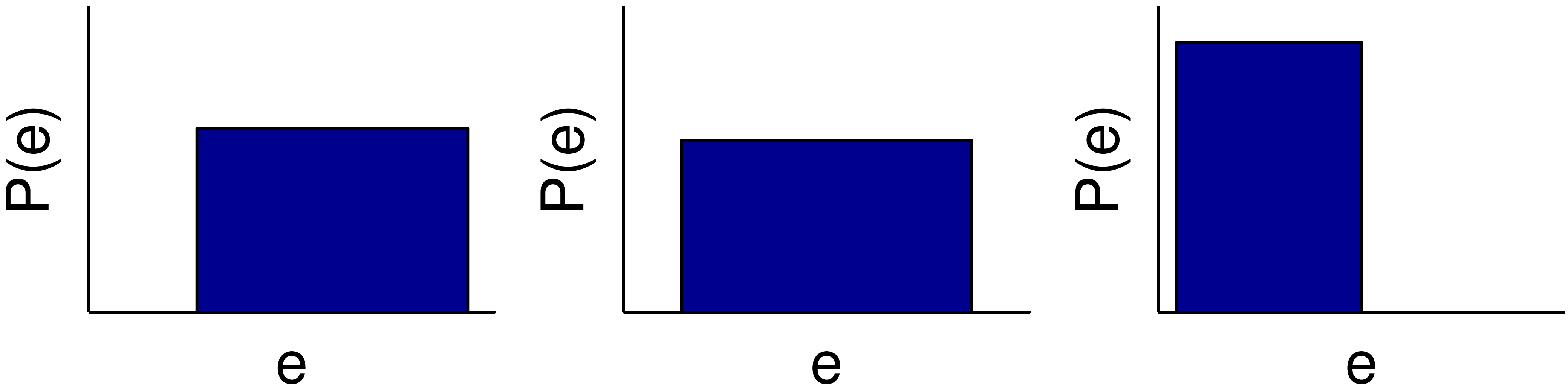
Hierarchical graphical model



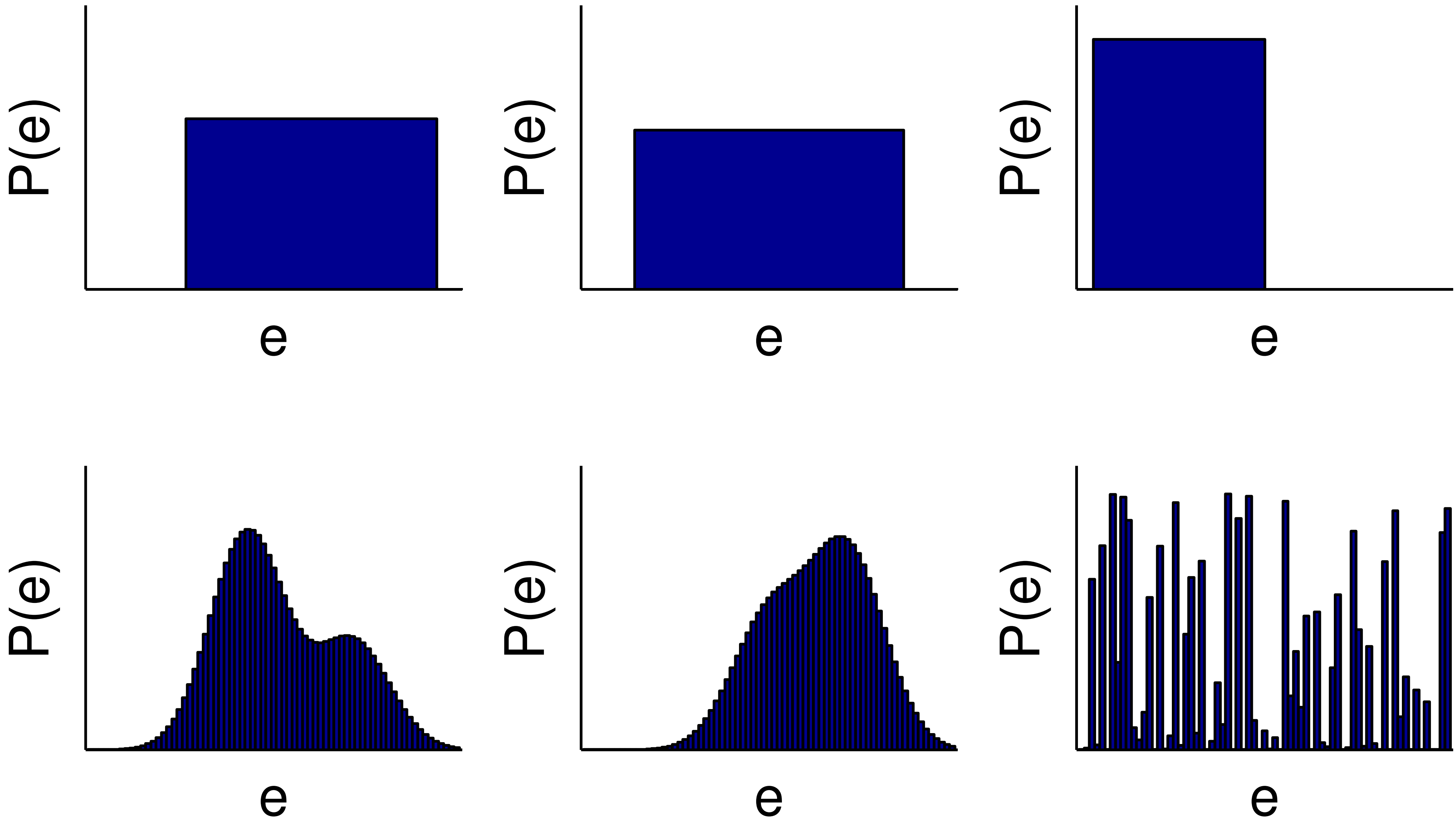
Inferences about the Sun



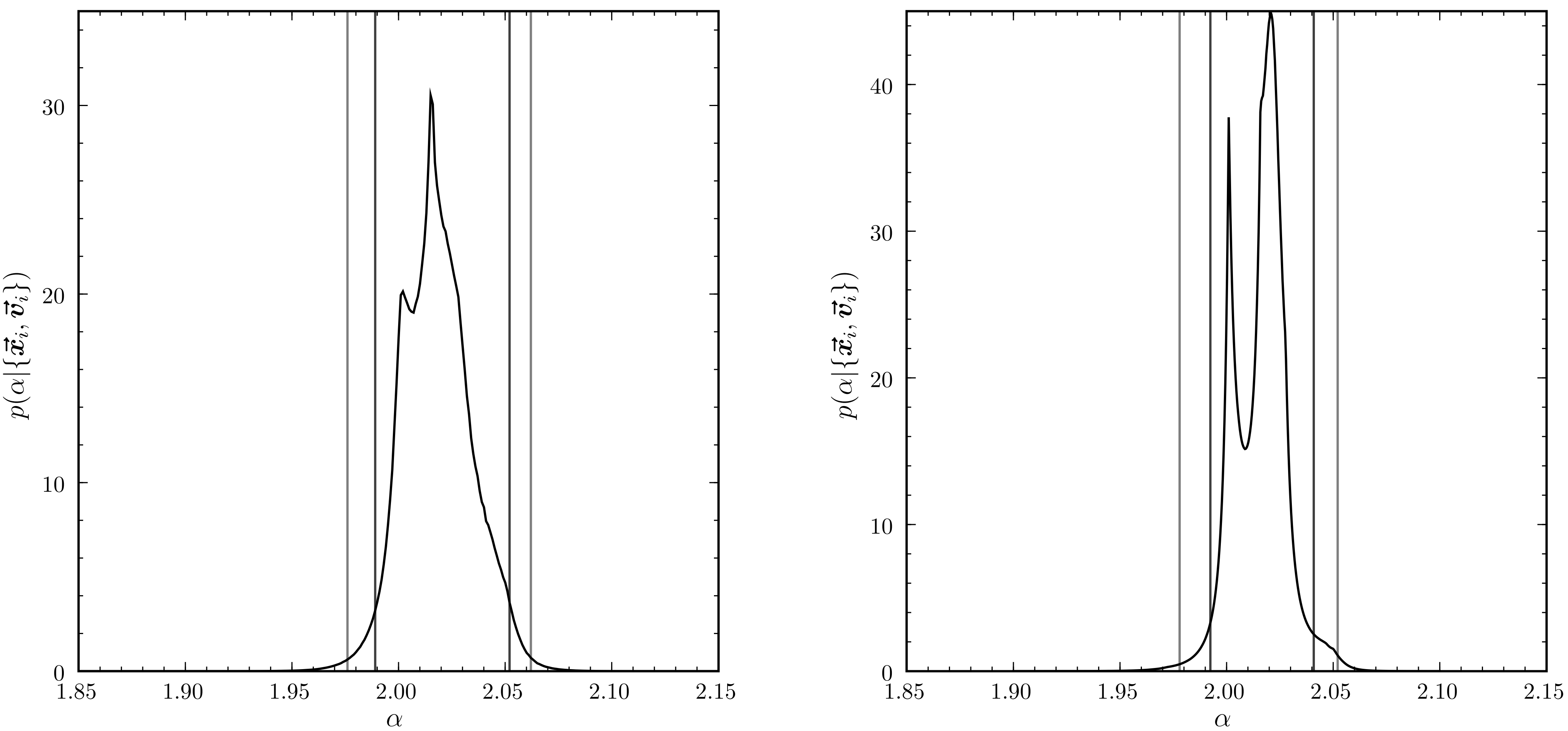
Priors on nuisance distributions

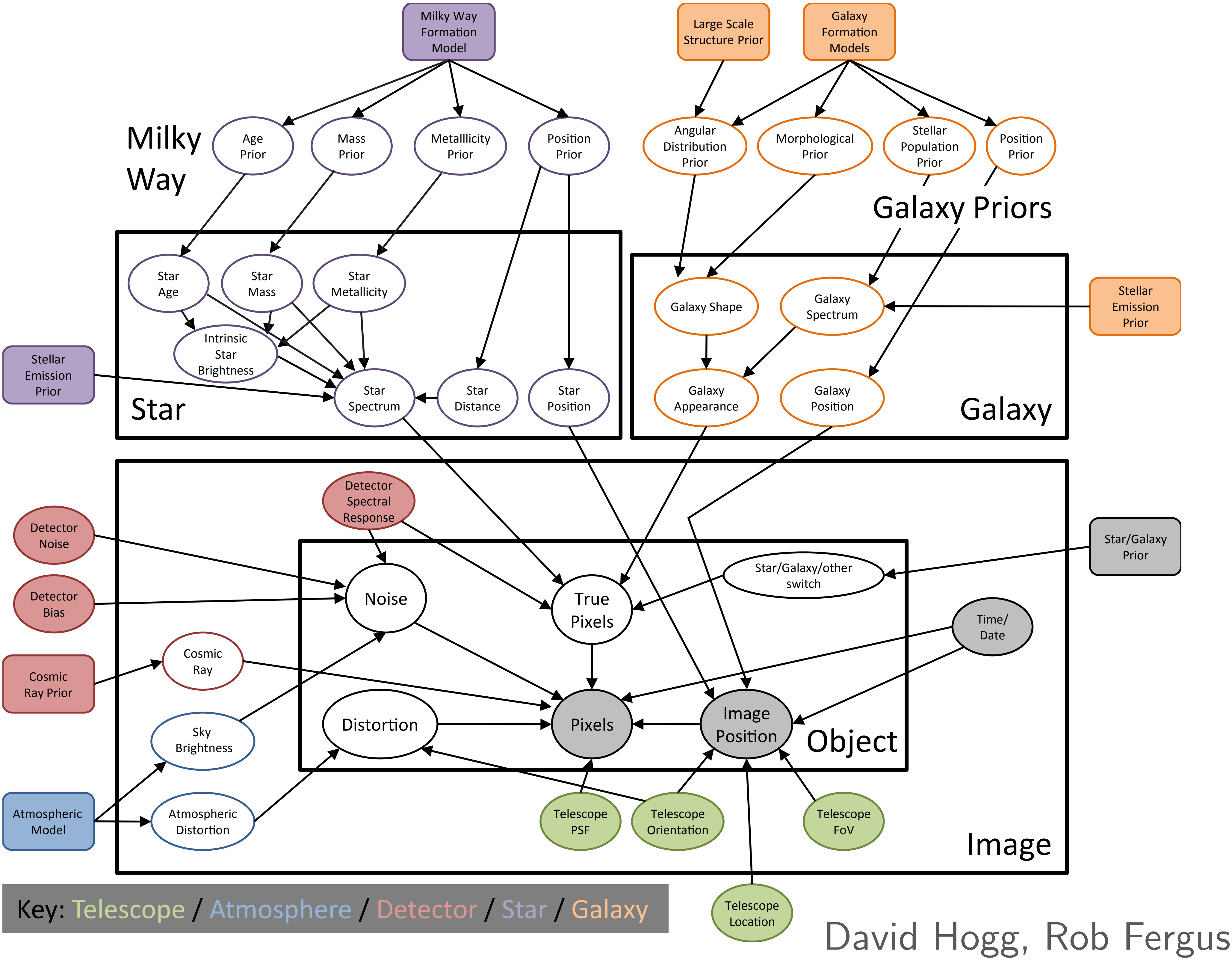


Priors on nuisance distributions



Gravitational exponent




Milky Way Formation Model
Large Scale Structure Prior
Galaxy Formation Models
Milky Way
Age Prior
Mass Prior
Metallicity Prior
Position Prior
Angular Distribution Prior
Morphological Prior
Stellar Population Prior
Position Prior
Galaxy Priors
Star Age
Star Mass
Star Metallicity
Intrinsic Star Brightness
Stellar Emission Prior
Star Spectrum
Star Distance
Star Position
Star
Galaxy Shape
Galaxy Spectrum
Stellar Emission Prior
Galaxy Appearance
Galaxy Position
Galaxy
Detector Spectral Response
Detector Noise
Detector Bias
Noise
True Pixels
Star/Galaxy/other switch
Star/Galaxy Prior
Time/ Date
Cosmic Ray
Cosmic Ray Prior
Sky Brightness
Distortion
Pixels
Image Position
Object
Atmospheric Model
Atmospheric Distortion
Telescope PSF
Telescope Orientation
Telescope FoV
Image
Telescope Location
Key: Telescope / Atmosphere / Detector / Star / Galaxy
David Hogg, Rob Fergus

Machine Learning?

— **Flexible probabilistic models**

Neural networks and Gaussian processes

— **Inference methods**

Statistical methods: MCMC, etc.

Learning: recognition networks

https://arxiv.org/abs/1605.06376

Computational tools

Backpropagating derivatives is ubiquitous

Statistical computations need more support

$$\theta \;\rightarrow\; \Sigma \;\rightarrow\; L \;\rightarrow\; \mathcal{F}$$

$$\frac{\partial \theta}{\partial \mathcal{F}} \;\leftarrow\; \frac{\partial \Sigma}{\partial \mathcal{F}} \;\leftarrow\; \frac{\partial L}{\partial \mathcal{F}} \;\leftarrow\; \mathcal{F}$$

First step: `http://arxiv.org/abs/1602.07527`

The End

Reusable tools for solving inference problems

Monte Carlo methods, Neural nets, and more

Derivative propagation for statistical computation